CLAIMS

What is claimed is:

1. A Web-based system that asynchronously processes synchronous requests, comprising:

an interface component that receives a synchronous request; and

a processing component that parses the synchronous request across a plurality of Web services for asynchronous processing, the processing component aggregates asynchronous results from the plurality of Web services and returns a synchronous result.

2. The system of claim 1, the processing component parses the synchronous request based on a load balancing technique that distributes portions of the synchronous request to one or more of the plurality of Web services so that request processing is spread across respective Web services based on Web service load.

3. The system of claim 2, the load balancing technique dynamically conveys portions of the request from a first Web service to a second Web service with a lesser load, during processing.

4. The system of claim 1, the parsed synchronous request is serially and/or concurrently processed by the plurality of Web services.

5. The system of claim 1, the synchronous request and result is conveyed across the interface component *via* at least one of the following protocols: TCP/IP; IPX/SPX; UDP/IP; HTTP; SOAP; and a proprietary protocol.

6. The system of claim 1 further comprises a queue that is utilized to post the synchronous request for retrieval by one or more Web services that are subscribed to process requests.

7. The system of claim 6, the queue is utilized to store information indicative of at least one of a querying client, the synchronous request, the interface component, the processing component, the processing component queue, and a connection type.

8. The system of claim 7, the information is utilized to track the request as it is being processed by the Web services, correlate results from the plurality of Web services with the synchronized request, and/or return a synchronous result.

9. The system of claim 1, the plurality of Web services comprise respective queues that store information indicative of at least one of a synchronous request provider, the synchronous request, the interface component, the processing component, the process component queue, a connectivity type, the Web service, and the Web service queue.

10. The system of claim 1 further comprises an API that facilitates conveyance of the received synchronous request to the processing component and conveyance of the synchronous result.

11. The system of claim 1 further comprises an error-handling component that transmits a message indicating processing of the request has been halted due to a time period lapse.

12. The system of claim 1 further comprises an error-handling component that facilitates re-distributing portions of the request from a Web service that is unable to process the portion to another Web service where the portion is processed.

13. The system of claim 1 further comprises an intelligence component that facilitates one or more of determining Web service load, parsing the request, distributing the parsed request, correlating results, grouping results, and returning synchronized result.

14. The system of claim 13, the intelligence component employs at least one of a statistic, a probability, an inference and a classifier.

15. A system that employs dynamic load balancing to asynchronously process synchronous requests, comprising:

a processing engine that posts synchronous requests in a message box that is accessed by one or more subscribed Web-based services that asynchronously process the synchronous requests;

an aggregating component that correlates asynchronous results with the synchronous request and groups the correlated results; and

an output component that returns the grouped results as a synchronous result.

16. The system of claim 15 further comprises an adapter that accepts a synchronous request from a client and conveys the synchronous request to the processing engine.

17. The system of claim 16, the adapter is one of a pluggable software component and an instance of an object.

18. The system of claim 15, the synchronous requests is delineated across the one or more subscribed Web-based services, based on a dynamic load balancing technique that distributes portions of the synchronous request to one or more of the subscribed Web-based services, according to Web-based service load.

19. The system of claim 15 is employed within one or more of an intranet, an internet and the Internet.

20. The system of claim 15 further comprises an error-handling component that provides a notification when the request cannot be serviced.

21. The system of claim 15, the processing engine facilitates re-distribution of portions of the synchronous request to one or more Web-based services based on load.

22. The system of claim 15, the message box is utilized to store information indicative of at least one of a querying client, the synchronous request and the message box.

23. The system of claim 22, the information is employed by the output component to facilitate returning the synchronous result.

24. A method that facilitates Web-based asynchronous processing of synchronous requests, comprising:

accepting a synchronous request;

dynamically delineating the synchronous request across process engines based on process engine load;

correlating asynchronous results and errors; and

returning the correlated results as a synchronous result.

25. The method of claim 24 further comprises publishing the synchronous request in a message queue.

26. The method of claim 24 further comprises subscribing process engines with the message queue.

27. The method of claim 24 further comprises notifying a requester when the request fails to be processed.

28. A method for asynchronously processing a synchronous request on a Web service, comprising:

transmitting a synchronous request;

distributing the synchronous request across severs within a server farm;

aggregating asynchronous results with an associated synchronous request; and

returning the aggregated results as a synchronous result.

29. The method of claim 28 further comprises utilizing request related information to facilitate one or more of tracking the synchronous request through processing, aggregating results, re-distributing portions of the request between servers, and returning a synchronous result to a client.

30. The method of claim 28 further comprises distributing the synchronous request in a dynamic manner based on server load.

31. The method of claim 28 further comprises serially and/or concurrently processing the synchronous request.

32. A data packet transmitted between two or more computer components that facilitates Web-based asynchronous processing of synchronous requests, comprising:

a component that receives a synchronous request from a client, a component that posts the request in a queue, a component that parses the stored synchronous request across servers within a farm of servers based on a dynamic balancing technique, a component that correlates asynchronous results with the synchronous request, and a component that returns the asynchronous results to the client as a synchronous result.

33. A computer readable medium storing computer executable components to dynamically processes a synchronous request *via* a Web service that utilizes load-based asynchronous processing, comprising:

a component that dynamically distributes the synchronous request across processing engines based on load;

a component that dynamically re-distributes the synchronous request as processing engine load changes;

a component that correlates asynchronous results with the synchronous request; and

a component that returns the asynchronous results as a synchronous result.

34. A Web-based system that employs dynamic asynchronous processing to service synchronous requests, comprising:

means for receiving a synchronous request;

means for posting the synchronous request;

means for dynamically distributing the synchronous request across processing engines based at least on process engine load;

means for correlating asynchronous results with the synchronous request; and

means for returning the asynchronous results as a synchronous result.